NASA Technical Memorandum 100217
ICOMP-87-5

Approximate Polynomial Preconditioning Applied to Biharmonic Equations on Vector Supercomputers

(NASA-TM-100217) APPROXIMATE POLYNOMIAL PRECONDITIONING APPLIED TO BIHARMONIC EQUATIONS ON VECTOR SUPERCOMPUTERS (NASA) 20 p Avail: NTIS HC A03/MF A01 CSCL 12A G3/64 N88-10563 Unclas 0104551

Yau Shu Wong
Institute for Computational Mechanics in Propulsion
Lewis Research Center
Cleveland, Ohio

and

Hong Jiang
University of Alberta
Edmonton, Alberta, Canada

October 1987

NASA

APPROXIMATE POLYNOMIAL PRECONDITIONING APPLIED TO BIHARMONIC EQUATIONS ON VECTOR SUPERCOMPUTERS

*Yau Shu Wong**
Department of Mathematics, University of Alberta, Edmonton, Alberta, CANADA,
and Institute for Computational Mechanics in Propulsion,
NASA Lewis Research Center, Cleveland, Ohio, USA

and

Hong Jiang†
Department of Mathematics, University of Alberta, Edmonton, Alberta, CANADA

SUMMARY

Applying a finite difference approximation to a biharmonic equation results in a very ill-conditioned system of equations. This paper examines the conjugate gradient method used in conjunction with the generalized and approximate polynomial preconditionings for solving such linear systems. An approximate polynomial preconditioning is introduced, and is shown to be more efficient than the generalized polynomial preconditionings. This new technique provides a simple but effective preconditioning polynomial, which is based on another coefficient matrix rather than the original matrix operator as commonly used.

INTRODUCTION

The conjugate gradient (CG) method used in conjunction with a suitable preconditioning has been widely accepted as one of the most efficient iterative procedures for solving large and sparse systems of linear equations. The basic CG algorithm can be efficiently implemented on a vector computer. However, the traditional preconditioning technique, which is generally based on an incomplete LU factorization procedure (Meijerink and van der Vorst, 1977) is not suitable for coding on a vector computer. The solution for the preconditioning step is obtained via forward and backward substitutions, which are difficult to vectorize because of the recursive nature. To overcome this problem, the use of a matrix polynomial as a preconditioner has been proposed (Dubois et al., 1979). The preconditioning operator is taken to be an approximation of the matrix inverse, which can then be obtained by taking sufficient number of terms in the matrix polynomial expansion. The main advantage of such an approach is that the preconditioning step can be implemented essentially via matrix by vector multiplications, which do not require forward and backward substitutions.

---

* *This work was supported in part by the Natural Sciences and Engineering Research Council of Canada Grant U0375; and in part by NASA (funded under the Space Act Agreement C99066G) while the author was visiting ICOMP, NASA Lewis Research Center.*

† *The work of this author was supported by an Izaak Walton Killiam Memorial Scholorship.*

A very important practical consideration in the preconditioned CG algorithm is that the preconditioning step should be easily computed. Notice that the computational work for each basic CG iteration essentially comprises of one matrix by vector multiplication, two vector inner products, and three vector additions. For the incomplete LU preconditioning procedure, the amount of work due to the preconditioning is about one matrix by vector multiplication. However, it becomes k matrix by vector multiplications when k terms are kept in the polynomial preconditioning technique. Hence, the computational work per CG iteration when using a polynomial preconditioning may be dominated by the cost required for the preconditioning step when k is greater than one. An earlier paper (Wong, 1987) presented an approximate polynomial preconditioning technique that may be more efficient than the traditional polynomial preconditionings. For matrix problems resulting from high-order discretization methods applied to elliptic partial differential equations, this method provides a substantial saving in computing time compared to that with the usual polynomial preconditioning. The basic idea of this technique is that the preconditioning polynomial is derived from another coefficient matrix that contains far fewer non-zero elements than the original matrix operator.

This paper demonstrates how an approximate polynomial preconditioning can be applied to matrix equations resulting from a biharmonic problem. Although our discussion will be restricted to symmetric and positive definite matrices, the extension to general non-symmetric matrices with positive definite symmetric part is straightforward.

RESULTS AND DISCUSSION

Consider the Dirichlet problem for the biharmonic equation in a square D with boundary ∂D

$$\Delta^2 u(x,y) = f(x,y) \qquad \text{in } D \qquad (1)$$

and

$$u(x,y) = g_1(x,y)$$

$$u_n(x,y) = g_2(x,y) \qquad \text{on } \partial D$$

where $\Delta^2 u = u_{xxxx} + 2u_{xxyy} + u_{yyyy}$, and u_n denotes the normal derivative of $u(x,y)$.

When Eq.(1) is discretized by a finite difference scheme with a uniform mesh h, the result is a system of linear equations

$$Bu = b \qquad (2)$$

where B is a symmetric matrix with coefficients (except those close to the boundary) given by the following 13-point finite difference stencil

$$
\begin{array}{ccccc}
 & & 1 & & \\
 & 2 & -8 & 2 & \\
1 & -8 & 20 & -8 & 1 \\
 & 2 & -8 & 2 & \\
 & & 1 & &
\end{array} \qquad (3)
$$

It should be pointed out that although B is a symmetric and positive definite matrix, it does not possess the property of diagonal dominance. Furthermore, B is very ill conditioned and has a condition number proportional to h^{-4}. Hence, the application of an iterative procedure, such as the CG algorithm, results in a very slow rate of convergence.

The basic preconditioned CG algorithm is outlined in Section 1, and the approximate polynomial preconditioning technique is described in Section 2. The calculation of parameters to improve the preconditioning polynomial is presented in Section 3. The choice of a special initial solution to accelerate the CG algorithm is discussed in Section 4. From the computational experiments given in Section 5, it appears that the use of the CG algorithm in conjunction with an approximate polynomial preconditioning results in a highly efficient iterative procedure for the solution of a biharmonic equation on a vector computer.

1. PRECONDITIONED CONJUGATE GRADIENT ALGORITHM

Let the preconditioning operator M be a non-singular matrix. Eq.(2) can be rewritten as

$$BM^{-1}Mu = b \qquad (4)$$

The CG algorithm applied to the preconditioned system (4) is summarized as follows:

Initialization:

Step 1. Choose an initial solution u^0

Step 2. Compute residual vector $r^0 = b - Bu^0$

Step 3. Set direction vector $p^0 = M^{-1}r^0$

Iteration: For $n=0,1,2,\ldots$, until convergence, do

Step 4. $\alpha_n = (r^n, M^{-1}r^n)/(p^n, Bp^n)$

Step 5. $u^{n+1} = u^n + \alpha_n p^n$

Step 6. $r^{n+1} = r^n - \alpha_n Bp^n$

Step 7. $\beta_n = (r^{n+1}, M^{-1}r^{n+1})/(r^n, M^{-1}r^n)$

Step 8. $p^{n+1} = M^{-1}r^{n+1} + \beta_n p^n$

The inner product is defined as $(x,y)=x^Ty$ for any vector x and y. When M=I, it becomes the basic CG algorithm applied to the original system (2).

The basic CG algorithm is suitable for coding on a vector computer such as the CDC CYBER 205. Notice that steps 5, 6, and 8 are the well-known linked triad (i.e., vector + constant • vector) operations, and the two inner products required in step 4 and 7 can be computed by using the Q8SDOT routine available on the CYBER 205 computer. The main computational work in each CG iteration is one matrix by vector multiplication Bp, which can be implemented almost entirely by the linked triad operations (Madsen et al., 1976). Consequently, the work for each preconditioned CG iteration is given by

$$W = W_b + W_p \tag{5}$$

where W_b is the work for the basic CG algorithm and

$$W_b = 2*IP + 3*LT + 1*MV \tag{6}$$

Here IP, LT, and MV denote operations for the inner product, linked triad, and matrix by vector multiplication. W_p is the work resulting from the preconditioning step, and $W_p=0$ if M=I.

It is well known that the convergence rate of the CG algorithm can be substantially improved by the application of a good preconditioning matrix M. The important considerations in the selection for M are that M^{-1} should be a good approximation to B^{-1} in some sense, and that the preconditioning step $M^{-1}r$ can be conveniently computed. A popular choice for M, which can be efficiently implemented on a vector computer, is based on a Neumann series for approximating a matrix inverse (Dubois et al., 1979). This procedure is generally referred to as a polynomial preconditioning technique.

Suppose the original matrix B has been symmetrically scaled so that $b_{ii}=1.0$ for all i, B can then be expressed as

$$B = I - G \tag{7}$$

where G is symmetric with $g_{ii} = 0.0$ for all i. Since $B^{-1}=(I-G)^{-1}$, M^{-1} can be obtained via a truncated matrix series expansion in G; that is,

$$M^{-1} = \sum_{i=0}^{k} G^i \tag{8}$$

For convergence, the spectral radius of G must be less than one.

Unfortunately, when B is resulting from a biharmonic equation, it can not be conveniently scaled in the form given in (7) while keeping G symmetric. Due to the boundary conditions, b_{ii} is not constant: it has a value of 20 for all i in the interior points, and it becomes 21 or 22 for i corresponding to the boundary points.

To alleviate this difficulty it is natural to generalize the polynomial preconditioning technique. One approach is by considering the matrix splitting of B. Let B=D–Q, where D is the diagonal matrix, then

$$M^{-1} = (\sum_{i=0}^{k} G^i)\, D^{-1} \qquad (9)$$

where $G = D^{-1}Q$. Notice that the preconditioning matrix M is symmetric, and the spectral radius of G is less the one if and only if D+Q is positive definite (Adams, 1985). Another approach is by expressing B is the following form:

$$B = \omega\,[I-(I-(B/\omega))] = \omega(I-G) \qquad (10)$$

where ω is a parameter. It is not hard to verify that if $\omega > \lambda max/2$ (where λmax is the largest eigenvalue of B), the spectral radius of G is less than one. A simple choice for ω is $\omega = ||B||/2$, where $||B||$ is taken to be the maximum row sums of the matrix B. Hence, $B^{-1}=\omega^{-1}(I-G)^{-1}$, and the generalized polynomial preconditioning matrix can be defined as

$$M^{-1} = \sum_{i=0}^{k} G^i = \sum_{i=0}^{k} (I-(B/\omega))^i \qquad (11)$$

The main advantage of this approach is that the original matrix B does not need to be symmetrically scaled. Furthermore, it can be applied to non-symmetric problems and matrices which are not diagonally dominant. It is easy to verify that $\omega=1$ for scaled matrices resulting from a finite difference approximation to Laplace operators. For these problems, the generalized polynomial preconditioning (11) reduces to the simple form given in (8).

2. APPROXIMATE POLYNOMIAL PRECONDITIONING

The generalized polynomial preconditionings described in the previous section can be expressed as

$$M^{-1} = (\sum_{i=0}^{k} G^i)\, N \qquad (12)$$

In order to avoid confusion between these two versions, the following abbreviations will be used:

(1) DPP_1, where $G=D^{-1}Q$, $N=D^{-1}$, and $B=D-Q$
(2) DPP_2, where $G=I-(B/\omega)$, $N=I$, $\omega=||B||/2$

Here, DPP denotes the direct polynomial preconditioning (i.e., the polynomial series is based on the original matrix).

Notice that M^{-1} becomes a good approximation to the matrix inverse B^{-1} as the number of terms k increases in the polynomial expansion (12). The preconditioning step $M^{-1}r$ required in each CG iteration can be calculated via

$$M^{-1}r = (I + G + G^2 + \ldots + G^k)\,\bar{r} \qquad (13)$$

where $\bar{r}=Nr$. The computational work due to the preconditioning is then essentially given by

$$W_p = k*LT + k*MV \qquad (14)$$

if k terms are kept in the preconditioning polynomial. Since G has the same structure as the original matrix B, the arithmetic operations for one matrix by vector multiplication Bp required in W_b (Eq.(6)) is almost the same as that for $G\bar{r}$ needed in W_p (Eq.(14)). The computational work for each CG iteration given in Eq.(5) may then be dominated by W_p if k is greater than one. Consequently, reduction in the cost for W_p will result in an overall improvement for the preconditioned CG iterative process.

An earlier paper (Wong, 1987) shows that for certain matrix problems it is possible to obtain an efficient polynomial preconditioning in which the polynomial expansion is based on another coefficient matrix that consists of far fewer non-zero elements than the original matrix operator. To distinguish from the traditional polynomial preconditioning, this new approach will be referred to as an approximate polynomial preconditioning technique.

Consider B as the matrix resulting from the difference approximation to a biharmonic equation. It can then be shown that

$$B = L^2 + E \qquad (15)$$

where L is the difference approximation to the Laplace operator, and E is a matrix with small rank. Hence, when solving the biharmonic equation $Bu=b$, a polynomial preconditioning can be considered in which the matrix polynomial is based on L rather than the original matrix B. The approximate

polynomial preconditioning (APP) can be expressed as

$$M^{-1} = \sum_{i=0}^{k} G^i \qquad (16)$$

where $G=I-(L/\omega)$. Since L is the Laplace operator, $\omega=1$, and $G=I-L$. The attractive feature in using APP instead of DPP_1 or DPP_2 is that the computational cost for W_p is substantially reduced when k is large. Notice that the cost for one matrix by vector multiplication $G\bar{r}$ based on APP is less than half of that of the DPP_1 or DPP_2 methods. This is due to the fact that for two dimensional problems, the matrix B has 13 non-zero diagonals, whereas L has 5 non-zero diagonals.

3. PARAMETER SELECTION

The basic generalized and approximate polynomial preconditionings have been described in the previous sections. More efficient polynomials, however, can be achieved by introducing appropriate parameters into the preconditioning polynomial. Several authors have discussed ways of finding the parameters. For more details see Johnson et al. (1983) and Saad (1985). This section briefly outlines how the parameters are determined to improve the approximate polynomial preconditioning.

By introducing the parameter γ_i into the APP given in (16),

$$M^{-1} = \sum_{i=0}^{k} \gamma_i G^i \qquad (17)$$

where $G=I-L$. The parameters γ_i are chosen so that the eigenvalue distribution of the preconditioned system $M^{-1}B$ is more favourable to the CG method. In general, this will also result in a smaller condition number for $M^{-1}B$. To satisfy these requirements, the choice of γ_i should minimize $||I-M^{-1}B||$. However, it is more convenient to consider $||I-M^{-1}L^2||$ by ignoring the contribution of E given in Eq.(15).

Recall that M^{-1} defined in Eq.(17) is a polynomial of degree k in G. If $M^{-1}=P_k(G)$, it is then clear that $M^{-1}L^2$ is a polynomial of degree $k+2$; hence,

$$M^{-1}L^2 = Q_{k+2}(G) = P_k(G)(I-G)^2 = \sum_{i=0}^{k} \gamma_i G^i (I-G)^2 \qquad (18)$$

Since the spectral radius of G is less than one, that is,

$$-1 < \lambda_0 \leq \ldots \leq \lambda_n < 1 \qquad (19)$$

where λ_j denotes an eigenvalue of G. The corresponding eigenvalues of $M^{-1}L^2$ are then given by

$$Q_{k+2}(\lambda_j) = \sum_{i=0}^{k} \gamma_i \lambda_j^i (1-\lambda_j)^2 \qquad (20)$$

A good choice for γ_i is to make the quadratic norm of

$$\int_{-1}^{+1} [Q_{k+2}(\lambda) - 1\,]^2 d\lambda \qquad (21)$$

as small as possible. Applying a least square method to minimize the integral (21), it leads to a normal equation

$$C\gamma = c \qquad (22)$$

where C is a dense matrix of order k+1. Alternatively, the method of orthogonal polynomial expansion can be applied to construct a three term recurrence relation for the preconditioning polynomial. The advantage of this approach is that it does not require solving the ill-conditioned normal equation. The following are the resulting first three polynomials:

$$k=1, \quad P_1(G) = I + (7/8)G$$
$$k=2, \quad P_2(G) = (49/40)I + (98/40)G + (63/40)G^2$$
$$k=3, \quad P_3(G) = (25/24)I + (72/24)G + (117/24)G^2 + (66/24)G^3$$

The same procedure can be applied to determine suitable parameters for the DPP_1 and DPP_2 preconditionings, which will not be described here.

4. SPECIAL INITIAL SOLUTION

An initial solution vector u^0 is required to start the CG iterative procedure. In general, the rate of convergence of the CG method is not sensitive to the initial guess u^0. It has been found that when the preconditioned CG algorithm is applied to the elliptic difference equations with $u^0=0.0$, or u^0 consisting of uniform random numbers, or $u^0=M^{-1}b$ where M^{-1} is a preconditioning polynomial which approximates the original matrix inverse, there is little difference in the numbers of iterations required for convergence (Wong, 1987).

For the solution of biharmonic equations $Bu=b$, however, it is possible to determine an initial vector u^0, which gives an improved convergence rate for the CG method. Recall that Eq.(15) stated that $B=L^2+E$. Now, ignoring the matrix E, u^0 can be obtained by

$$L^2u^0=b \qquad (23)$$

The solution for u^0 can be found via solving

$$\begin{aligned} Ly &= b \\ \text{and} \qquad & \qquad (24) \\ Lu^0 &= y \end{aligned}$$

Since L is the Laplacian difference matrix, the solutions for Eq.(24) can be computed very efficiently by the CG method with a polynomial preconditioning. The question to be asked now is: Whether this choice for u^0 provides a good starting initial solution for the CG method?

Suppose u is the exact solution of the biharmonic equations, and let

$$u^0 = u + e \qquad (25)$$

The initial residual vector is then given by

$$r^0 = b - Bu^0 = b - B(u+e) = -Be \qquad (26)$$

From Eq.(25),

$$\begin{aligned} Bu^0 &= Bu + Be \\ (L^2+E)u^0 &= b + Be \qquad (27) \\ Eu^0 &= Be \end{aligned}$$

Hence, $$r^0 = -Eu^0 \qquad (28)$$

Therefore, u^0 provides a small initial residual as long as the norm of E is small. However, it should be pointed out that this is not the only factor which results in a better rate of convergence for the CG method. The initial solution u^0 also has a special property that leads to improved convergence rates. Notice that u^0 is the solution of $L^2u^0=b$, and it is well known that the eigenfunctions of the Laplace and biharmonic operators are closely related (Birkhoff and Lynch, 1984). Consequently, not only that the resulting initial residual vector is small, but it also contains components of the eigenvectors of the biharmonic operator. A faster convergence rate can therefore be achieved when the CG method is used with this particular choice for u^0. This remark is verified by the computational experiments.

5. COMPUTATIONAL RESULTS

In order to determine the effectiveness of the approximate polynomial preconditioning, consider the solution of a biharmonic equation (1) over a square region with $0 \leq x,y \leq 1$. The functions $g_1(x,y)$ and $g_2(x,y)$ are chosen so that the exact solution is given by

$$u(x,y) = x^2y^2(x-1.0)^2(y-1.0)^2 \qquad (29)$$

Eq.(1) is discretized on a uniform mesh h, and the total number of equations is given by $EQ=n^2$, where $h=1/(n+1)$. Numerical experiments have been performed on the CDC CYBER 205 computer with 2-pipe and 64-bit arithmetic. For the results reported in this section, the CG iterative process is terminated when the maximum $|r_i|$ is less than 10^{-10}, where r is the residual vector.

5.1 DIRECT VERSUS APPROXIMATE POLYNOMIAL PRECONDITIONINGS

The performance of the preconditioned CG algorithm with various preconditionings is examined for $h=1/100$. The initial solution vector u^0 is composed of random numbers uniformly distributed in [0,1]. In Table 1, results for DPP_1, $DPP2_2$, and APP preconditionings are compared, where k is the number of terms in the polynomials. Note that no parameter is introduced into the polynomial preconditionings.

Table 1. Number of iterations for EQ=9801

(No parameter in preconditionings)

k	0	1	2	3	4
DPP_1	5512	2777	4260	2683	4054
DPP_2	5492	2777	3162	1954	2439
APP	5492	2192	3206	1690	2504

From the results reported in Table 1, it is clear that a large number of iterations is required for convergence when the CG method is used without preconditioning (i.e., $k=0$). It is also observed that DPP_2 provides better performance than the DPP_1 preconditioning. Hence, our attention for direct polynomial preconditionings will be focused on DPP_2, and it will hereafter simply be referred to as DPP.

As previously mentioned, more efficient preconditioning can be obtained by introducing appropriate parameters into the polynomial. The effect of the parameters γ_i on DPP and APP preconditionings is shown in Table 2. The parameters are calculated by the least square method as described in Section 3. The corresponding CPU time in seconds is given in Figure 1.

Table 2. Number of iterations for EQ=9801

(Parameterized preconditionings)

k	0	1	2	3	4	5	10	20	30
DPP	5492	3024	2239	1712	1416	1203	685	379	261
APP	5492	2335	1448	1002	778	623	299	142	97

Several interesting phenomena can be seen in these results. First, parameterized preconditionings provide continuous improvement as the number of terms k increases unlike those based on the basic polynomial preconditionings, as is seen in Table 1. By comparing the results corresponding to k=5, 10, 20, and 30, it is noted that the number of iterations required for convergence is reduced by half as k increases by a factor of two. Secondly, the APP preconditioning should be preferred. Not only is the number of iterations for a fixed value of k is smaller for the APP than for the DPP method, but the computing time is also significantly smaller. The standard CG method with no preconditioning requires 18.5 seconds for convergence, whereas the best results for the CG+DPP and CG+APP methods are 13.1 and 2.2 seconds, respectively.

The effectiveness of the polynomial preconditionings can be best illustrated by examining the eigenvalue spectrum of BM^{-1}. Figure 2 gives the original spectrum of the biharmonic operator (i.e., M=I) for n=16 (i.e., EQ=225). In Figures 3 and 4, we show the corresponding eigenvalue spectrum of the preconditioned operator using DPP and APP preconditionings for k=1, 5, 10, and 15. The condition numbers $C=\lambda max/\lambda min$ are also reported in each case, where λmax and λmin are the largest and smallest eigenvalues. In Figures 2 to 4, each eigenvalue is plotted by a vertical bar. From these figures, it is seen that when the number of terms k increases for both DPP and APP preconditionings, not only are the condition numbers reduced but the eigenvalues are also more clustered and thus more favourably distributed for the CG method.

It is evident that of the two preconditionings considered here, the APP method is the most effective for the biharmonic problems. This observation is somewhat unexpected, because it might be thought that the use of the DPP method (which is based on the original matrix operator) would give a better eigenvalue spectrum for the CG method, and hence would require a smaller number of iterations. However, it may not be as efficient as the APP because of the large amount of work would be needed to perform the preconditioning steps. Numerical results given in Figures 3 and 4, on the other hand, reveal that for a given number of terms k in the preconditioning polynomials, the

condition number for the preconditioned system based on the APP method is actually smaller than that based on the DPP method. Consequently, the APP preconditioning results in a faster rate of convergence for the CG iteration, as is reported in Table 2. Moreover, since the computational work for each preconditioned CG iteration based on the APP method is much smaller than that for the DPP preconditioning, a substantial improvement in the performance of the CG method is achieved.

5.2 EFFECT ON SPECIAL INITIAL SOLUTION

Table 3 presents the results of the DPP and APP methods for EQ=9801, in which the initial solution is obtained via solving $L^2u^0=b$ as described in Section 4. The corresponding CPU time in seconds is shown in Figure 5, where the time includes that for finding the initial guess u^0.

Table 3. Number of iterations for EQ=9801

(Parameterized preconditionings and special u^0)

k	0	1	2	3	4	5	10	20	30
DPP	1379	758	561	430	356	303	174	95	66
APP	1379	596	370	262	207	169	88	47	33

The results presented in Table 3 and Figure 5 are compared to those of Table 2 and Figure 1, which are based on an initial solution where u^0 consists of random numbers in [0,1]. The results clearly illustrate the advantage of using the special choice of u^0, namely their ability to substantially reduce the number of iterations required for convergence. Consequently, a significant saving in computing time is achieved. The best results for solving EQ=9801 for DPP and APP methods are 3.8 and 1.2 seconds compared to 13.2 and 2.2 as reported earlier.

5.3 COMPARISON OF CG AND CG+APP METHODS

The performance of the standard CG without preconditioning and that with APP preconditioning together with the special initial starting solution u^0 is now investigated. Table 4 compares the number of iterations required for convergence for various numbers of equations. For the APP method, k is fixed to be 25 for all cases tested here. The corresponding CPU time is also given in Figure 6, where CG+APP is the total time required to solve Bu=b and the time taken to compute $u^0=(L^2)^{-1}b$.

Table 4. Number of iterations

n	100	150	200	250
EQ	9801	22201	39601	62001
CG	5492	11674	19568	31462
CG+APP	32	71	126	170

These results speak for themselves. The standard CG method is not practical for solving ill-conditioned biharmonic equations due to their very slow convergence rates. However, significant improvement is clearly achieved when it is applied in conjunction with the APP technique and with the special choice for u^0. For EQ=62001, the computing time for CG+APP is 28.5 seconds, whereas the CG method requires 686.1 seconds; this represents a reduction by a factor of 24.

6. CONCLUSIONS

Many practical problems in elasticity and fluid dynamics require the solution of equations involving biharmonic operators. Discretizing the equation by a finite difference method results in a very ill-conditioned system of linear equations. The computational experiments reported in this paper show that the conjugate gradient method with no preconditioning is not practical because of its slow convergence rates. However, the convergence rate can be substantially improved by the use of a preconditioning technique. Of the two versions of the generalized polynomial preconditionings DPP_1 and DPP_2 presented in this paper, the DPP_2 method is preferred. This method provides a better performance than the DPP_1 method, and it does not require the matrix to be symmetrically scaled or to possess the property of diagonal dominance.

Another technique introduced, namely, the approximate polynomial preconditioning (APP), provides a significant improvement over the traditional polynomial preconditioning. The conclusion favouring the APP method is based largely on the grounds of computational work involved in the preconditioning step. The application of the CG algorithm in conjunction with the APP technique results in a highly efficient iterative procedure for the solution of biharmonic equations. The method is also suitable for implementation on vector supercomputers. Notice that the APP and DPP_2 methods can also be applied with the generalized CG method (Axelsson 1980 and Eisentat et al. 1983) for non-symmetric matrix problems. Extension of the present method for general fourth order elliptic partial differential equations and practical problems of interest to computational fluid mechanics is under investigation; the results will be reported in a separate paper.

REFERENCES

1. Adams, L. 1985. M-step preconditioned conjugate gradient methods. SIAM J. Sci. Stat. Comp. 6(2):452-463.

2. Axelsson, O. 1980. Conjugate gradient type methods for unsymmetric and inconsistent systems of linear equations. Lin. Alg. Appl. 29:1-16.

3. Birkhoff, G., and Lynch, R.E. 1984. Numerical solution of elliptic problems. Philadelphia: SIAM.

4. Dubois, M., Greenbaum, A., and Rodrigue, G. 1979. Approximate the inverse of a matrix for use in iterative algorithms on vector processors. Computing 22:257-268.

5. Eisentat, S.C., Elman, H.C., and Schultz, M.H. 1983. Variational iterative methods for non-symmetric systems of linear equations. SIAM J. Numer. Anal. 20(2):345-357.

6. Johnson, O.G., Micchelli, C.A., and Paul, G. 1983. Polynomial preconditioners for conjugate gradient calculations. SIAM J. Numer. Anal. 20(2):362-376.

7. Madsen, N., Rodrigue, G., and Karush, J. 1976. Matrix multiplication by diagonals on a vector parallel processor. Inf. Proc. Letts. 5:41-45.

8. Meijerink, J.A., and van der Vorst, H.A. 1977. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. Math. Comp. 31:148-162.

9. Saad, Y. 1985. Practical use of polynomial preconditionings for the conjugate gradient method. SIAM J. Sci. Stat. Comp. 6(4):865-881.

10. Wong, Y.S. 1987. Solving large elliptic difference equations on CYBER 205. Parallel Computing (to appear).

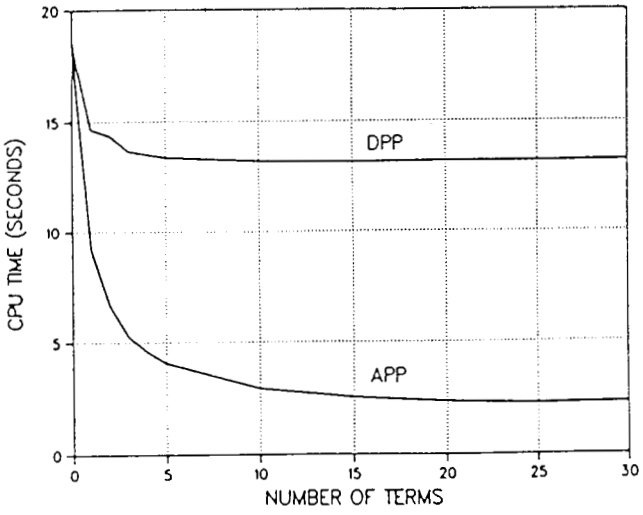


Fig. 1 Comparison of two preconditionings for EQ=9801

(initial u^0 based on random numbers)

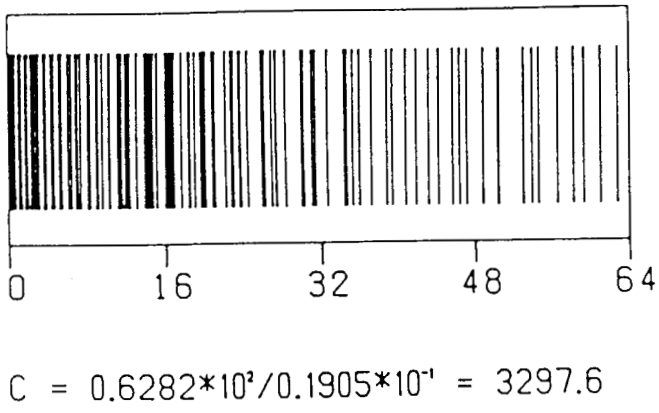


Fig. 2 Eigenvalue spectrum of the original matrix B for EQ=225

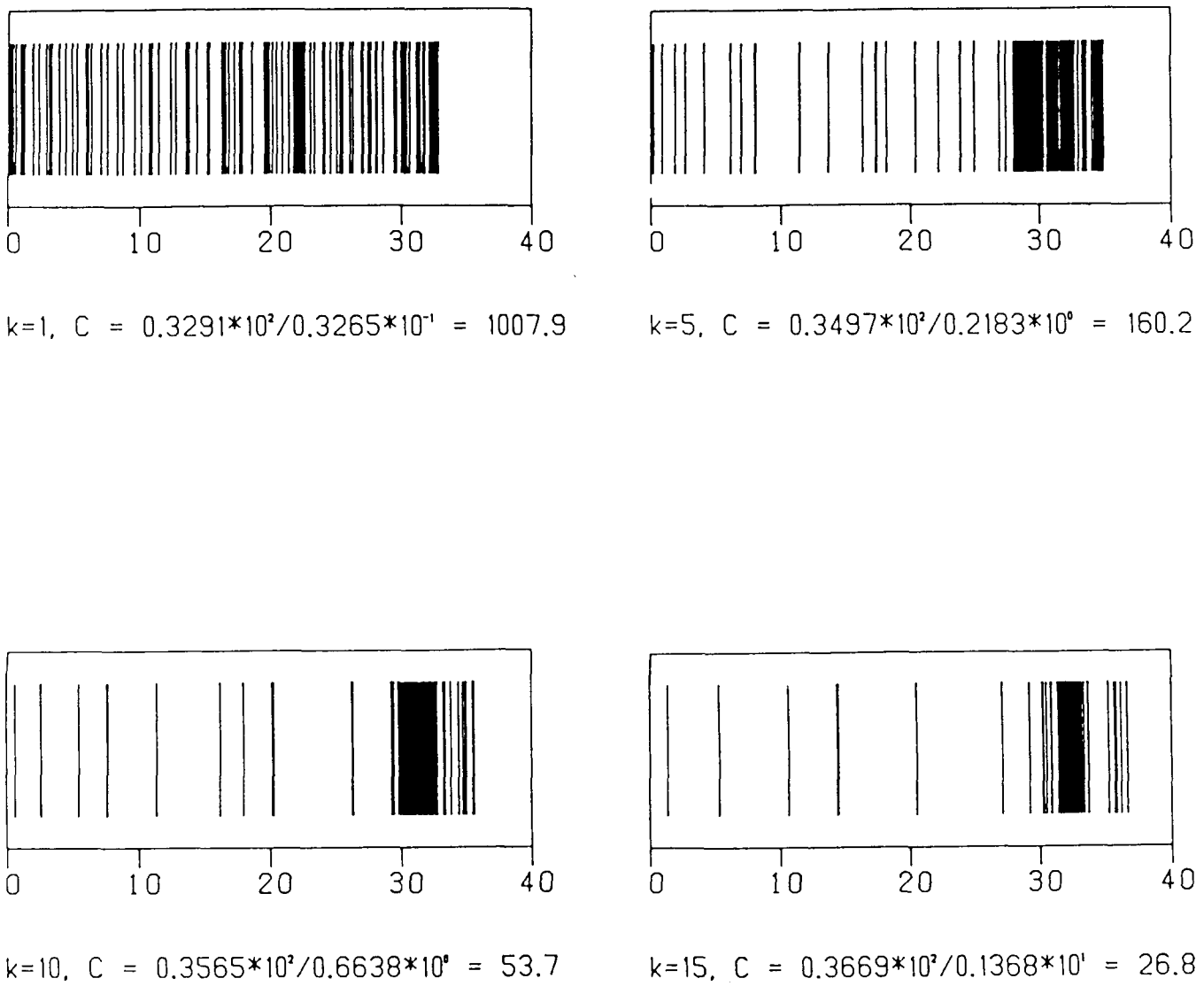
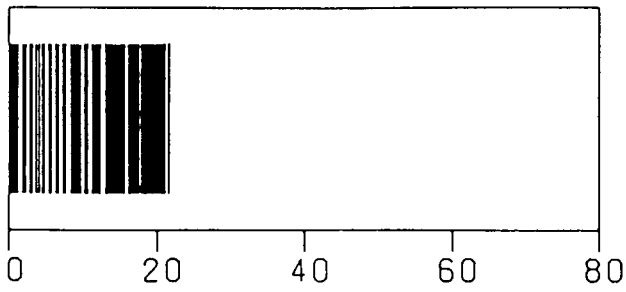
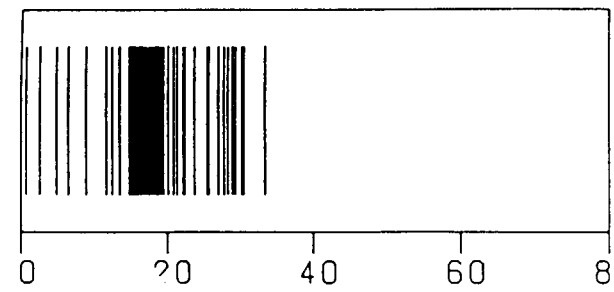


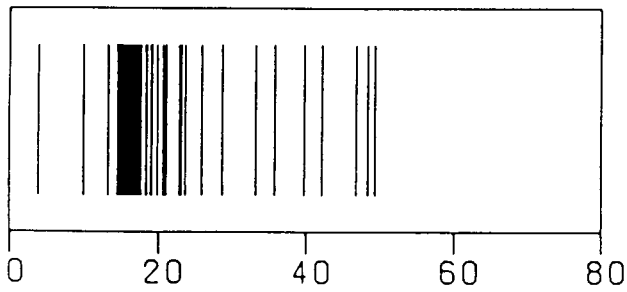
Fig. 3 Eigenvalue spectra of the preconditioned matrices BM^{-1} for EQ=225 (DPP method for k=1, 5, 10, and 15)



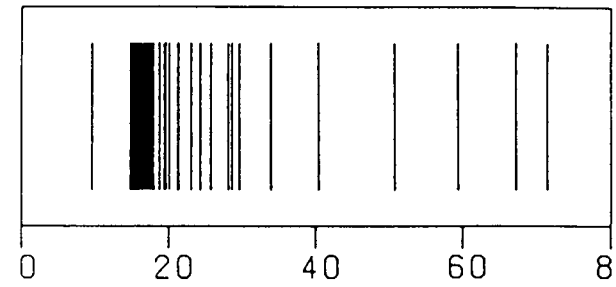
k=1, C = $0.2164*10^{2}/0.3533*10^{-1}$ = 612.5



k=5, C = $0.3344*10^{2}/0.7137*10^{0}$ = 46.9



k=10, C = $0.4935*10^{2}/0.3884*10^{1}$ = 12.7



k=15, C = $0.7147*10^{2}/0.9526*10^{1}$ = 7.5

Fig. 4 Eigenvalue spectra of the preconditioned matrices BM^{-1} for EQ=225 (APP method for k=1, 5, 10, and 15)

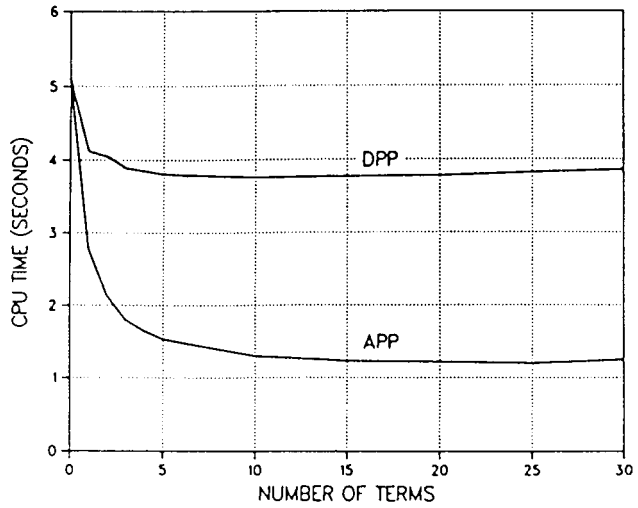


Fig. 5 Comparison of two preconditionings for EQ=9801 (initial $u^0=(L^2)^{-1}b$)

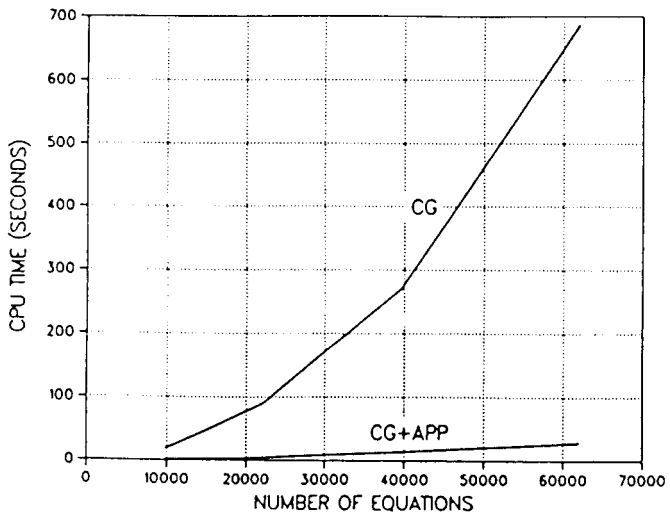


Fig. 6 Comparison of CG and CG+APP methods

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-100217 ICOMP-87-5		

4. Title and Subtitle

Approximate Polynomial Preconditioning Applied to Biharmonic Equations on Vector Supercomputers

5. Report Date

October 1987

6. Performing Organization Code

7. Author(s)

Yau Shu Wong and Hong Jiang

8. Performing Organization Report No.

E-3826

10. Work Unit No.

505-62-21

9. Performing Organization Name and Address

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135-3191

11. Contract or Grant No.

13. Type of Report and Period Covered

Technical Memorandum

12. Sponsoring Agency Name and Address

National Aeronautics and Space Administration
Washington, D.C. 20546-0001

14. Sponsoring Agency Code

15. Supplementary Notes

Yau Shu Wong, Department of Mathematics, University of Alberta, Edmonton, Alberta, Canada (work funded in part by the Natural Sciences and Engineering Research Council of Canada Grant U0375), and the Institute for Computational Mechanics in Propulsion, NASA Lewis Research Center (work funded under Space Act Agreement C99066G). Hong Jiang, Department of Mathematics, University of Alberta, Edmonton, Alberta, Canada.

16. Abstract

Applying a finite difference approximation to a biharmonic equation results in a very ill-conditioned system of equations. This paper examines the conjugate gradient method used in conjunction with the generalized and approximate polynomial preconditionings for solving such linear systems. An approximate polynomial preconditioning is introduced, and is shown to be more efficient than the generalized polynomial preconditionings. This new technique provides a simple but effective preconditioning polynomial, which is based on another coefficient matrix rather than the original matrix operator as commonly used.

17. Key Words (Suggested by Author(s))

Polynomial preconditioning
Biharmonic equations

18. Distribution Statement

Unclassified – Unlimited
Subject Category 64

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No of pages	22. Price*
Unclassified	Unclassified	19	A02

*For sale by the National Technical Information Service, Springfield, Virginia 22161